

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-201182

(P2000-201182A)

(43) 公開日 平成12年7月18日 (2000.7.18)

(51) Int.Cl.<sup>7</sup>

識別記号

F I

テーマコード (参考)

H 0 4 L 12/56

H 0 4 L 11/20

1 0 2 D

5 K 0 3 0

12/46

11/00

3 1 0 C

5 K 0 3 3

12/28

9 A 0 0 1

審査請求 未請求 請求項の数14 O L (全 10 頁)

(21) 出願番号 特願平11-3201

(22) 出願日 平成11年1月8日 (1999.1.8)

特許法第30条第1項適用申請有り 1998年11月20日 社団法人電子情報通信学会発行の「電子情報通信学会技術研究報告 信学技報 vol. 98 No. 408」に発表

(71) 出願人 000004226

日本電信電話株式会社

東京都千代田区大手町二丁目3番1号

(72) 発明者 安川 正祥

東京都新宿区西新宿三丁目19番2号 日本電信電話株式会社内

(72) 発明者 山中 直明

東京都新宿区西新宿三丁目19番2号 日本電信電話株式会社内

(74) 代理人 100078237

弁理士 井出 直孝 (外1名)

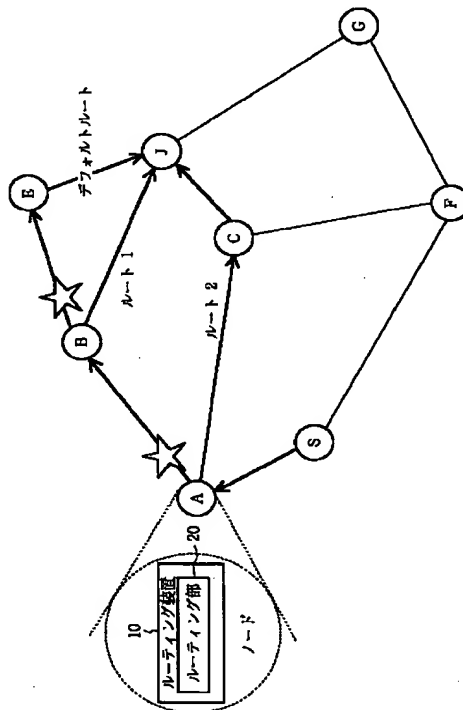
最終頁に続く

(54) 【発明の名称】 ルーティング装置およびルーティング方法

(57) 【要約】

【課題】 高速かつ高信頼の packets 転送を達成する。

【解決手段】 ネットワーク全体に周期的に配信され、各ノードが同期して保持するリンクステート情報と各ノードが局所的に管理するインタフェース情報をもとに任意のインタフェース内に輻輳が発生した場合に当該インタフェースを避けてアダプティブにルーティングパスを設定する。



## 【特許請求の範囲】

【請求項 1】 ネットワーク内で同期して分配されるグローバルな IP アドレスとノードの位置関係を記述するノードトポロジー情報と各ノード毎に管理されノード内のインタフェースを通じて転送されるトラヒック負荷を反映したローカルな輻輳情報とにしたがって輻輳ポイントを避けたリルーティングパスを任意の宛先に対してループフリーに設定する手段を備えることを特徴とするリルーティング装置。

【請求項 2】 前記ループフリーに設定する手段は、ループフリーなリルーティングパスの方向を示すルーティングベクトル情報を保持する手段を含む請求項 1 記載のルーティング装置。

【請求項 3】 ループフリーなリルーティングパスの中で最小メトリックを備えるパス順にリルーティング経路を設定する手段を備える請求項 1 記載のルーティング装置。

【請求項 4】 最小メトリックパスおよびそれ以外の複数のリルーティングパスのトラヒック負荷の情報を保持する手段と、この保持する手段に保持された情報にしたがって目的宛先までネットワーク内での負荷分散を行いながらルーティングを実行する手段を備える請求項 1 記載のルーティング装置。

【請求項 5】 前記ルーティングを実行する手段は、リンクステート交換周期よりも小さい一定周期毎に、実際に転送されるトラヒック負荷を反映した転送コストにより前記トラヒック負荷の情報を更新する手段を含む請求項 4 記載のルーティング装置。

【請求項 6】 階層化された IP ネットワーク内における階層間をまたがってループフリーでマルチパスを設定し負荷分散を行いながらルーティングを実行する階層間ルーティング実行手段を備える請求項 1 記載のルーティング装置。

【請求項 7】 前記階層間ルーティング実行手段は、自階層および自階層よりも上位または下位の階層のトラヒック負荷および輻輳の情報を管理してルーティングを実行する代表ノードを備える請求項 6 記載のルーティング装置。

【請求項 8】 ネットワーク内で同期して分配されるグローバルな IP アドレスとノードの位置関係を記述するノードトポロジー情報と各ノード毎に管理されノード内のインタフェースを通じて転送されるトラヒック負荷を反映したローカルな輻輳情報とにしたがって輻輳ポイントを避けたリルーティングパスを任意の宛先に対してループフリーに設定することを特徴とするルーティング方法。

【請求項 9】 ループフリーなリルーティングパスの方向を示すルーティングベクトル情報にしたがいリルーティングパスを任意の宛先に対してループフリーに設定する請求項 8 記載のルーティング方法。

【請求項 10】 ループフリーなリルーティングパスの中で最小メトリックを備えるパス順にリルーティング経路を設定する請求項 8 記載のルーティング方法。

【請求項 11】 最小メトリックパスおよびそれ以外の複数のリルーティングパスのトラヒック負荷の情報にしたがって目的宛先までネットワーク内での負荷分散を行いながらルーティングを実行する請求項 8 記載のルーティング方法。

【請求項 12】 リンクステート交換周期よりも小さい一定周期毎に、実際に転送されるトラヒック負荷を反映した転送コストにより前記トラヒック負荷の情報を更新する請求項 11 記載のルーティング方法。

【請求項 13】 階層化された IP ネットワーク内における階層間をまたがってループフリーでマルチパスを設定し負荷分散を行いながらルーティングを実行する請求項 8 記載のルーティング方法。

【請求項 14】 自階層および自階層よりも上位または下位の階層のトラヒック負荷および輻輳の情報を管理してルーティングを実行する請求項 13 記載のルーティング方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は IP ネットワーク内の各ノード（ルータ）に搭載するルーティングアルゴリズムに利用する。本発明はダイナミックに変動するトラヒック環境下でアダプティブに経路変更を行い輻輳ポイントを回避したルーティングを行うことにより高速かつ高信頼の packets 転送を可能とするルーティングアルゴリズムに使用する。特にネットワークが階層化された大規模な IP ネットワークで輻輳ポイントを回避してマルチパスを設定し負荷分散を行いながらルーティングを行うスケーラブルな packets 転送技術に関する。

## 【0002】

【従来の技術】現在インターネットの世界で広く利用されているルーティングアルゴリズムは大きく 2 つのタイプのプロトコルに分類される。そのうちの一つはリンクステート型プロトコルと呼ばれるもので OSPF がその代表的な例である。このルーティングプロトコルでは、各ノードが隣合うノードまでのリンクステート情報（リンクコスト）をネットワーク全体に同期（ネットワーク内の各ノードの保持するデータが完全に一致）して配信する必要がある。

【0003】このとき、ネットワーク内の各ノードは配信されたリンクステート情報をもとにネットワークトポロジーを計算し、ネットワーク内の全ノードがネットワークトポロジーを反映した同一の有効グラフを共有する。各ノードはこの経路情報をもとに最小メトリック（距離）を達成するルートを検索することで目的宛先までの最短ルートを計算し各宛先毎のネクストホップノードを確定する。この過程では各ルータが完全に同期した

## 3

有効グラフを用いて最短ルートの計算を行うためにホップバイホップでパケットを転送しても各ノードが転送するネクストホップノードは最短ルート上に一致して存在する。このため同一の最短ルートを用いてパケットを目的宛先まで転送することが可能である。

【0004】二つ目の代表的なルーティングプロトコルがパスペクトル型のルーティングプロトコルである。BGPがその代表的な例である。パスペクトル型のルーティングプロトコルではネットワーク内の各ノードは隣接するノードまでの転送コストを隣接ノードに通知する。通知を受けた隣接ノードは自分自身の隣接ノード迄の転送コストを加算して次に隣接する隣接ノードまでトータルのホップコストを通知する。この操作をネットワーク全体に波及して行くと、各ノードは任意の宛先までのホップコストと隣接するネクストホップノードアドレスを決定することができる。

## 【0005】

【発明が解決しようとする課題】リンクステート情報をもとにルーティング経路を決定するリンクステート型アルゴリズムでは各ルータが保持する有効グラフが同期していることを前提としている。そのため各ルータが同期していない有効グラフをもとに最短ルートを計算すると各ノード間で計算する最短ルートの不一致が生じパケット転送時にループを形成してしまう問題が存在する。

【0006】一方、高スループットかつ高信頼のルーティングを行うためにはダイナミックに変動するネットワーク内の輻輳ポイントを抑えたアダプティブなルーティングが必要となる。このためネットワーク内で局所的に、しかもダイナミックに変動するトラヒック状況をネットワークに同期して配信するメトリックに反映させて各ノードがルーティング経路を計算することが望ましい。

【0007】しかしながらネットワーク規模が大きくなると、トラヒックの変動周期よりもメトリック配信周期の方が大きくなるので、各ルータ間でダイナミックに変動するトラヒック負荷を反映したメトリックを計算しネットワーク全体に同期して配信することは不可能となる。そのため負荷に対応したアダプティブな最短ルートを探索することは困難である。さらに、このアルゴリズムでは各ノード間のホップコストを反映したメトリック情報をもとにして最小メトリックルートから最短ルートを計算するため、計算される任意のノードから宛先ノードまでのルートは最小コストルートのみとなる。したがってネットワークの負荷状態に関わらず唯一のパケット転送ルートを用いてルーティングを行うために、ネットワーク内リソースを反映した負荷分散を意識したマルチパスルーティングが実現できない問題が存在する。

【0008】また、パスペクトル型のルーティングプロトコルでもルーティングパスを決定する際にはネットワーク全体に同期して転送コストが配信されることを前提

## 4

とするために、先に説明したリンクステート型のプロトコルと同様に局所的に発生した輻輳ポイントを抑えたアダプティブなルーティングが実現できない問題点が存在する。

【0009】さらに、パスペクトル型のルーティングプロトコルを用いた場合にもAS間のルーティングを行う場合にAS内のトラヒック状況を考慮したルーティングを行うことが困難なので階層化したネットワーク間では固定ルートを選択してルーティングを行っている。このためルート内のAS内で輻輳が発生した場合にダイナミックに迂回ルートを設定不能となりトラヒック集中ポイントでパケット廃棄が多発しネットワーク全体のスループット特性を著しく劣化させる問題点が存在する。

【0010】このため、階層化した大規模IPネットワークで高負荷時にアダプティブに迂回ルートを設定しマルチパス環境下でネットワーク全体の負荷分散を実行できるスケラブルなルーティングアルゴリズムが必要となる。

【0011】本発明は、このような背景に行われたものであり、ネットワーク内でダイナミックに変動して発生する輻輳ポイントを抑えたアダプティブなルーティングを実行することができるルーティング装置およびルーティング方法を提供することを目的とする。本発明は、ループフリーのルーティングを実行することができるルーティング装置およびルーティング方法を提供することを目的とする。本発明は、マルチパス環境下での負荷分散を達成しルーティングを実行することができるルーティング装置およびルーティング方法を提供することを目的とする。本発明は、階層化されたIPネットワーク内においても輻輳ポイントを抑えたループフリーのマルチパスのルーティングが可能となるルーティング装置およびルーティング方法を提供することを目的とする。本発明は、大規模階層化された複雑なIPネットワークで高スループットかつ高信頼のパケット転送を実現することができるルーティング装置およびルーティング方法を提供することを目的とする。

## 【0012】

【課題を解決するための手段】本発明は、ネットワーク全体に周期的に配信され、各ノードが同期して保持するリンクステート情報と各ノードが局所的に管理するインタフェース情報をもとに任意のインタフェース内に輻輳が発生した場合に当該インタフェースを避けてアダプティブにルーティングパスを設定できることを主要な特徴とする。

【0013】従来の技術とは、局所情報をもとにアダプティブにルーティングを行ってもループフリーのルーティング経路を設定できること、マルチパス環境下で負荷分散を行いながらパケット転送を行なえること、階層化されたIPネットワーク内でも輻輳ポイントを抑えたループフリーのアダプティブなマルチパスルーティング

が可能などころが異なる。

【0014】すなわち、本発明の第一の観点はルーティング装置であって、ネットワーク内で同期して分配されるグローバルなIPアドレスとノードの位置関係を記述するノードトポロジ情報と各ノード毎に管理されノード内のインタフェースを通じて転送されるトラヒック負荷を反映したローカルな輻輳情報とにしたがって輻輳ポイントを選けたリルーティングパスを任意の宛先に対してループフリーに設定する手段を備えることを特徴とする。

【0015】前記ループフリーに設定する手段は、ループフリーなリルーティングパスの方向を示すルーティングベクトル情報を保持する手段を含むことが望ましい。これにより、ループを形成してしまう可能性のあるリルーティングパスの設定を回避することができる。

【0016】また、ループフリーなリルーティングパスの中で最小メトリックを備えるパス順にリルーティング経路を設定する手段を備えることが望ましい。

【0017】さらに、最小メトリックパスおよびそれ以外の複数のリルーティングパスのトラヒック負荷の情報を保持する手段と、この保持する手段に保持された情報にしたがって目的宛先までネットワーク内での負荷分散を行いながらルーティングを実行する手段を備えることが望ましい。これにより、輻輳の発生確率を低減させることができる。

【0018】このとき、前記ルーティングを実行する手段は、リンクステート交換周期よりも小さい一定周期毎に、実際に転送されるトラヒック負荷を反映した転送コストにより前記トラヒック負荷の情報を更新する手段を含むことが望ましい。これにより、最新のトラヒック負荷の情報にしたがって、負荷分散を行いながらルーティングを実行することができる。

【0019】また、階層化されたIPネットワーク内における階層間をまたがってループフリーでマルチパスを設定し負荷分散を行いながらルーティングを実行する階層間ルーティング実行手段を備えることが望ましい。これにより、階層化されたIPネットワークにおいても本発明ルーティング装置のルーティングアルゴリズムを用いることができる。

【0020】このとき、前記階層間ルーティング実行手段は、自階層および自階層よりも上位または下位の階層のトラヒック負荷および輻輳の情報を管理してルーティングを実行する代表ノードを備えることが望ましい。この代表ノードにより、各階層間に本発明ルーティング装置のルーティングアルゴリズムを用いることができる。

【0021】本発明の第二の観点はルーティング方法であって、ネットワーク内で同期して分配されるグローバルなIPアドレスとノードの位置関係を記述するノードトポロジ情報と各ノード毎に管理されノード内のインタフェースを通じて転送されるトラヒック負荷を反映し

たローカルな輻輳情報とにしたがって輻輳ポイントを選けたリルーティングパスを任意の宛先に対してループフリーに設定することを特徴とする。

【0022】このとき、ループフリーなリルーティングパスの方向を示すルーティングベクトル情報にしたがいリルーティングパスを任意の宛先に対してループフリーに設定することが望ましい。また、ループフリーなリルーティングパスの中で最小メトリックを備えるパス順にリルーティング経路を設定することが望ましい。

10 【0023】最小メトリックパスおよびそれ以外の複数のリルーティングパスのトラヒック負荷の情報にしたがって目的宛先までネットワーク内での負荷分散を行いながらルーティングを実行することが望ましい。このとき、リンクステート交換周期よりも小さい一定周期毎に、実際に転送されるトラヒック負荷を反映した転送コストにより前記トラヒック負荷の情報を更新することが望ましい。

【0024】階層化されたIPネットワーク内における階層間をまたがってループフリーでマルチパスを設定し負荷分散を行いながらルーティングを実行することもできる。このとき、自階層および自階層よりも上位または下位の階層のトラヒック負荷および輻輳の情報を管理してルーティングを実行することが望ましい。

【0025】

【発明の実施の形態】発明の実施の形態を図1を参照して説明する。図1は本発明ルーティング装置の要部ブロック構成図を含むルーティングイメージを示す図である。

【0026】本発明はルーティング装置10であって、  
20 ネットワーク内で同期して分配されるグローバルなIPアドレスとノードの位置関係を記述するノードトポロジ情報と各ノード毎に管理されノード内のインタフェースを通じて転送されるトラヒック負荷を反映したローカルな輻輳情報とにしたがって輻輳ポイントを選けたリルーティングパスを任意の宛先に対してループフリーに設定する手段であるルーティング部20を備えることを特徴とする。

【0027】ルーティング部20は、ループフリーなリルーティングパスの方向を示すルーティングベクトル情報を保持する。また、ループフリーなリルーティングパスの中で最小メトリックを備えるパス順にリルーティング経路を設定する。さらに、ルーティング部20は、最小メトリックパスおよびそれ以外の複数のリルーティングパスのトラヒック負荷の情報を保持し、この保持された情報にしたがって目的宛先までネットワーク内での負荷分散を行いながらルーティングを実行する。このとき、ルーティング部20は、リンクステート交換周期よりも小さい一定周期毎に、実際に転送されるトラヒック負荷を反映した転送コストにより前記トラヒック負荷の情報を更新する。  
50

【0028】また、本発明のルーティング装置10は、階層化されたIPネットワーク内における階層間をまたがってループフリーでマルチパスを設定し負荷分散を行いながらルーティングを実行する。このとき、自階層および自階層よりも上位または下位の階層のトラフィック負荷および輻輳の情報をルーティング部20により管理してルーティングを実行する代表ノードを備える。

【0029】

【実施例】提案するAMR (Adaptive Multipath Routing) アルゴリズムは各ノードがローカルに管理できるインタフェースコスト  $ds_k(\rho)$  とネットワーク全体が同期して管理する最短ルートコスト  $D_{kj}$  を用いてルーティングを行う。各ノードが管理するインタフェースコストは

$$ds_k(\rho) = ds_k(0) + IF(\rho)$$

で与えられる。ここで  $ds_k(0)$  は予めリンクステート情報交換時にネットワーク全体で同期して配信されるネクストホップコストで低負荷時にノード  $s$  からネクストホップノード  $k$  までのリンクコストを表わす。

【0030】図2はインタフェースコスト ( $IF-cost$ ) を示す図であり、横軸に負荷  $\rho$  をとり、縦軸にインタフェースコスト  $ds_k(\rho)$  をとると、 $IF(\rho)$  はノード  $s$  が管理するネクストホップノードまでのインタフェースコストを表し、図2に示すようにインタフェースに流入する負荷の関数となる。このコストは高負荷 ( $\rho > \rho_{th}$ ) 時には無限大に発散する。このようなネクストホップまでの輻輳情報を反映したコスト関数  $ds_k(\rho)$  と予めネットワークに同期して配信されるネクストホップから宛先までのコスト関数  $D_{kj}$  を用いて各ノードは最短コスト計算を行いルーティングを実行する。

【0031】FOR ALL POSSIBLE ROUTE

CALCULATE IF COST

$IFC[k] \leftarrow ds_k(\rho) + D_{kj}$

SELECT  $k$  THAT MINIMIZE IF COST

Next hop  $\leftarrow k$

このルーティングアルゴリズムを用いたルーティングイメージを図1に示す。図1はソースノード  $S$  から宛先ノード  $J$  迄のルーティング例である。ネットワークに同期して配信されるメトリックをもとに計算される最短ルートは  $S \rightarrow A \rightarrow B \rightarrow E \rightarrow J$  であるが、ノード  $B$  のインタフェース  $BE$  に輻輳が発生して一部のトラフィックがノード  $B$  から直接  $J$  に向かう例と、ノード  $A$  のインタフェース  $AB$  に輻輳が発生してトラフィックの一部を  $A \rightarrow C \rightarrow J$  と迂回させる例とを示している。このように各ノードがローカルな  $IF$  コストを計算し輻輳発生時にはアダプティブに迂回路を設定するためネットワーク全体で高スループットかつ高信頼のルーティングが可能となる。

(ループフリールーティング) 本発明のルーティングアルゴリズムでは各ノードがローカルなインタフェースコストをもとに自律分散的に輻輳を回避しながらルーティングを行うために輻輳ポイントを持つ複数のノードを通過した後で同一宛先を目指すパケットがループを形成する可能性が存在する。図3にノード  $C$  とノード  $F$  に接続されるリンクに輻輳が発生し、この輻輳ポイントを避けるために宛先  $J$  のパケットがループ  $A \rightarrow C \rightarrow F \rightarrow S$  を形成する例を示す。このようなルーティングループの形成を防止するために本発明アルゴリズムではネットワーク内の各ノードが同期してルーティングベクトルを計算する。ルーティングベクトルは十分長い周期でネットワーク全体に同期して配信されるリンクステート情報から計算され、宛先毎にネットワーク内でルーティング時に許容される方向ベクトルをあらわしている。

【0032】したがって、輻輳回避時にこのルーティングベクトルに一致した方向にリルート経路を設定すればループを形成しないことを保証することができる。この方向ベクトルは任意の宛先  $J$  を起点として  $J$  に到達する最短ルートを検出する逆方向の  $Dijkstra$  のアルゴリズムを用いて計算される。図4にこの計算手法を用いて計算したルーティングベクトルを示す。この例では宛先  $J$  をめざすルーティングベクトルをあらわしている。各ノードが同期してこの情報を保持している。図4におけるノード  $A, B, C, S, E, F, G, J$  間に記載された数字は、各ノード間の転送コストを反映したメトリックを示す。また、符号  $P1, P2, P3, P4, P5$  は、ノード  $J$  からのメトリックが小さい順に各ノードまでのパスをそれぞれ示す。

【0033】例えばノード  $A$  からは宛先  $J$  に向かうのにループフリーのルートは1)  $A \rightarrow B \rightarrow E \rightarrow J$ 、2)  $A \rightarrow B \rightarrow J$ 、3)  $A \rightarrow C \rightarrow J$  の3種類のルートが存在することを示している。また、この計算手法によって計算されるルーティングベクトルを各ノードで独立して記述するために各ノードは隣接するノードまでのルーティングフラグを設定する。

【0034】図4には併せてノード  $A, B$  が保持する宛先  $J$  までの  $flag$  情報を示す。フラグ値設定に当たってはパケット転送方路とルーティングベクトルが一致する場合には  $flag \leftarrow 0$ 、一致しない場合には  $flag \leftarrow \infty$  を設定する。例えばノード  $A$  が保持するフラグ情報は隣接ノード  $B, C$  迄は  $A \rightarrow B : 0, A \rightarrow C : 0$ 、 $S$  迄はループを形成する可能性があるので  $A \rightarrow S : \infty$  と設定する。このように逆方向の  $Dijkstra$  を用いた計算手法を用いれば任意の宛先  $J$  を起点にして最短ホップノードを構成する隣接ノードを順次決定していくのでネットワークベクトルにはループが形成されないことを保証する。

【0035】また、逆方向の  $Dijkstra$  法を用いれば、最短パスを検索する上では同一のアルゴリズムと

なるため、宛先 J までの最短ルート候補が必ず包含されて計算されるのでこのルートベクトルにはネットワーク内の任意のノード S から宛先 J までの最短パスが必ず包含されることになる。これを比較するために順方向の最短ルート計算結果を図 5 に示す。図 5 におけるノード A、B、C、SD、E、F、G、J 間に記載された数字は、各ノード間の転送コストを反映したメトリックを示す。また、符号 P1、P2、P3、P4、P5、P6 は、ノード SD からメトリックが小さい順に各ノードまでのパスをそれぞれ示す。また、( ) 内の数字はノード SD から各ノードまでの累積メトリックを示す。図 5 では、ノード SD が各宛先ノードを目指す packets を転送するとき使用するネクストホップノード情報を併せて示す。

【0036】これにより各ノードがこのネットワークベクトルを用いて最短ルートからの迂回路を形成しても迂回に伴ってループが形成されず、迂回した隣接ノードからは最小メトリックコストを持つルートにしたがってリルーティングされるので、リルーティングを実行してもデフォルトの最小コストに近いルートを通じて目的宛先までルーティングされることを保証している。

(本発明アルゴリズム) 本発明アルゴリズムはリンクステートプロトコルを基本とするルーティングアルゴリズムである。したがって AS (Autonomous System: 同一の管理者によって自動的にルーティングされる範囲) 内のネットワークトポロジーを把握するために各ノードはリンクステート情報を交換しノード間で完全に同期した有効グラフを形成する。本発明のアルゴリズムはこの有効グラフをもとに (1) 自ノードと (2) 隣接ノードから各宛先迄の最短ルートと (3) 宛先までの迂回路を設定するルーティングベクトルを計算し、その結果を図 1 で説明したルーティング部 20 に格納する。また、データベースを設け、ルーティングベクトルの計算結果をこのデータベースに格納しておき、各ノードに配信したり、各ノードからの要求にしたがって各ノードに転送するようにしてもよい。下記アルゴリズム 1. が各ノードが計算するリンク情報を記述する。

【0037】次に本発明アルゴリズムを実装したノードに packets が到着した場合の処理をアルゴリズム 2. 以下に記述し、ノード内の packet フローの概念図を図 6 に示す。ノードに packets が到着すると packet ヘッダ内の IP アドレスから目的ホストに到着するための同一 AS 内の目的ノードを決定する。このとき、本発明アルゴリズムでは目的ノードに到達できるマルチルートのパスを選択しているのでノードで実測された転送コストを反映した分配率でマルチパスの中から転送パスを選択する (例えばコネクション毎に)。その後マルチパスで転送コストを反映した負荷分散が図れるように packets を転送する。

【0038】本発明アルゴリズムではマルチパスの候補

を一定周期  $\Delta T$  ( $\ll$  リンクステート交換周期) 毎に実際に転送されるトラフィック負荷を反映した転送コストで更新する。本発明アルゴリズムでは  $\Delta T$  内の各インタフェースコストをモニタしてデフォルトルートのコストと比較する。この過程では各インタフェースコストの中から最小コストを持つものを選択する。選択されたインタフェースがデフォルトの最短パス上のインタフェースであればこのインタフェースには輻輳が発生していないことになるのでこのインタフェースを用いて packets を転送する。このとき、デフォルトルート以外に転送ルートが存在する場合には転送ルート候補から削除する。選択されたインタフェースがデフォルトの最短ルートのインタフェースと異なるときには選択されたインタフェースをマルチルート転送の候補に加える。この場合は先に述べたコスト関数の定義によりデフォルトのインタフェースに輻輳が発生しているので、デフォルトルートからトラフィックを規定の分配率で迂回させる。

#### 1. SET DATABASE

FOR ALL DESTINATIONS

20 Calculate Shortest-Path  
(Source i  $\rightarrow$  Destination j)

Calculate Shortest-Path  
(Next-hop k  $\rightarrow$  Destination j)

Set Routing-Vector (j)

#### 2. SELECT OPTIMAL ROUTE

FOR EACH PACKET COMMING

(ADDRESS RESOLUTION)

30 Dcs ID  $\leftarrow$  Address Resolution  
(IP Address)

(SELECT PATH AMONG POSSIBLE PATHS)

SP  $\leftarrow$  Select Path (Possible Path [ ])

(DISTRIBUTE PACKET)

Distribute Packet (SP)

(SELECT POSSIBLE PATH)

AFTER SEVERAL INTERVAL OBSERVATION

40 FOR ALL DESTINATION

Default Route  $\leftarrow$  Shortest-Path (Source i  $\rightarrow$  Destination j)

Select Route  $\leftarrow$  Select Min Route (j)

IF Select Route EQUAL TO Default Route

Next hop  $\leftarrow$  Default Route

Delete Extra-Route (Possible Path [ ]) ELSE IF Select

50 Route DIFFER FROM Default

```

Route
Next hop ← Multi Route
Add Next-Route (Possible Path [])
Select Min Route (Possible Path [])
Select Min Route (Des ID)
FOR ALL POSSIBLE ROUTE
CALCULATE ROUTE COST
RC[k] ← IFC[k] + flag
← djk(ρ) + Dkj + flag
SELECT k THAT MINIMIZE ROUTE COST
Min hop ← k

```

このような機構を用いることにより本発明アルゴリズムはネットワーク内の輻輳ポイントを回避してマルチルートで負荷分散を行いながらパケットを転送する。このため高スループットで低損失かつ低遅延のルーティングが可能となる。

(マルチパス階層化ルーティング) 次に、本発明アルゴリズムを階層化されたネットワークに適用する場合を考える。図7は階層化されたIPネットワークを示している。この例ではネットワークはLevel 1～Level 13までの3つの階層構造を持つ。

【0039】Level 3の階層では4つのノード1.0.0～4.0.0が相互に接続されるネットワークを構成する。この階層下にはレベル2のノードが存在し、それぞれ(1.1.0～1.3.0)～(4.1.0～4.4.0)のノードが存在し、図7に示すネットワークを構成する。レベル2の配下にも同様にレベル1のノードが存在し図7に示すネットワークを構成する。最上位のレベル3のネットワークでルーティングを行うためにレベル2のノード内で代表ルータが決定される。この代表ルータはレベル3階層のルーティング情報を保持し、レベル3のルーティング処理を行う。

【0040】図7の例ではネットワーク1.0.0内のレベル2のノード1.2.0、1.3.0に存在するノードがレベル3の代表ノードとなりレベル3のルーティング処理を担当する。以下同様に代表ルータが決定される。このようにk階層で代表ノードに選出されたノードはk+1、k階層のルーティング情報を管理しk+1、k階層のルーティングの処理を行う。代表ノード以外のノードはk階層内のみのルーティング情報を管理してk階層内に閉じたルーティング処理だけを担当する。このような階層化を行うことで各ノードが保持するルーティング情報を圧縮し各階層内のルーティング処理を高速化できる。また各階層に設置された代表ルータは上位階層のルーティング処理を行うために代表ルータ同士でリンクステート情報を交換する。

【0041】図8は階層化されたIPネットワークを示

している。図8の例ではレベルkのルーティングを行うためにレベルk+1のノード(1.2.0, 1.3.0)、(2.2.0, 2.3.0)、(3.1.0)、(4.1.0, 4.3.0)が代表ノードに選択されている。各レベル内では発明アルゴリズムが独立に動作しており同一レベル内で輻輳が発生するとアダプティブにマルチパスを設定して輻輳ポイントを回避する。また、代表ノードはレベルkのルーティングを管理しているのでレベルk内で輻輳が発生するとレベルk内で迂回路を設定してマルチパスで輻輳ポイントを迂回する。このとき、同一のAS内で宛先ノードJに到達可能な別の代表ノードが存在する場合には当該ノードを迂回路に設定する。また、このとき、先に述べたルーティングベクトルを設定して代表ノード間ではループを構成しないように仮定しておく。

【0042】図8を用いてノード1.1.0がノード3.2.0を目指す場合のパケット転送を説明する。ノード1.1.0はパケットの宛先から同一AS内に目的ノードが存在しないことを判断してデフォルトで設定されている代表ルータ1.3.0にパケットを転送する。ノード1.3.0は代表ノードなのでレベルkのルーティング情報を用いてパケットをノード2.0.0に転送する。ノード2.1.0は最短パスを用いてノード2.3.0にパケットを転送する。このとき、ノード2.3.0のインタフェースには輻輳が発生しているのでノード2.3.0はレベルkのルーティング情報を用いてノード2.2.0を経由してノード4.0.0にトラヒックの一部を転送する。ノード4.1.0はレベルkのルーティング情報をもとにノード4.3.0を経由してノード3.0.0に転送されノード3.1.0が3.2.0に転送する。このような転送プロトコルを用いるため、提案プロトコルは階層化したIP網でマルチパスのルーティングが可能となる。

#### 【0043】

【発明の効果】以上説明したように、本発明によれば、ネットワーク内でダイナミックに変動して発生する輻輳ポイントを避けたアダプティブなループフリーのルーティングが実行できる。この過程では負荷分散を達成するためにマルチパス環境でルーティングが可能である。さらに、本発明のルーティングアルゴリズムで提案される制御方法はネットワークの階層化に対してスケーラビリティを持つために階層化されたIPネットワーク内においても輻輳ポイントを避けたループフリーのマルチパスのルーティングが可能となる。この結果、本ルーティングアルゴリズムを用いれば大規模階層化された複雑なIPネットワークで高スループットかつ高信頼のパケット転送を実現できる。

#### 【図面の簡単な説明】

【図1】本発明ルーティング装置の要部ブロック構成図を含むルーティングイメージを示す図。

10

20

30

40

50

【図 2】 インタフェースコストを示す図。

【図 3】 ノード C とノード F に接続されるリンクに輻輳が発生し、この輻輳ポイントを避けるために宛先 J のパケットがループ A→C→F→S を形成する例を示す図。

【図 4】 ルーティングベクトルを示す図。

【図 5】 順方向の最短ルート計算結果を示す図。

【図 6】 ノード内のパケットフローの概念図。

【図 7】 階層化された IP ネットワークを示す図。

【図 8】 階層化された IP ネットワークを示す図。

【符号の説明】

10 ルーティング装置

20 ルーティング部

A、B、C、E、F、G、J、S、SD、1. 0. 0、

2. 0. 0、3. 0. 0、4. 0. 0、1. 1. 0、

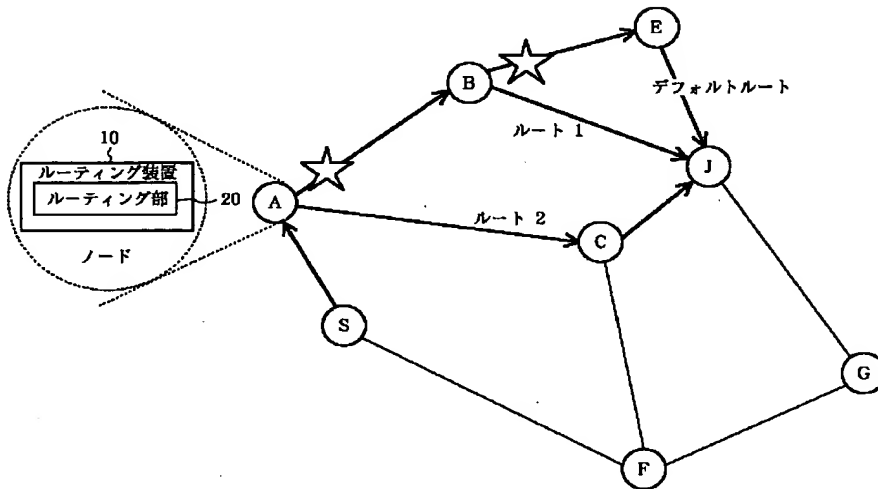
1. 2. 0、1. 3. 0、2. 1. 0、2. 2. 0、

2. 3. 0、3. 1. 0、3. 2. 0、4. 1. 0、

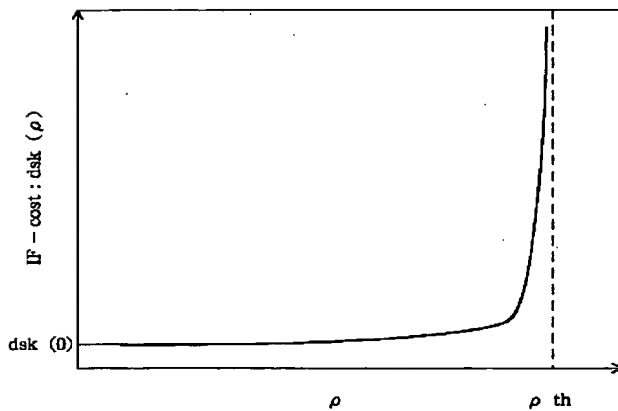
4. 2. 0、4. 3. 0、4. 4. 0 ノード

P1、P2、P3、P4、P5、P6 パス

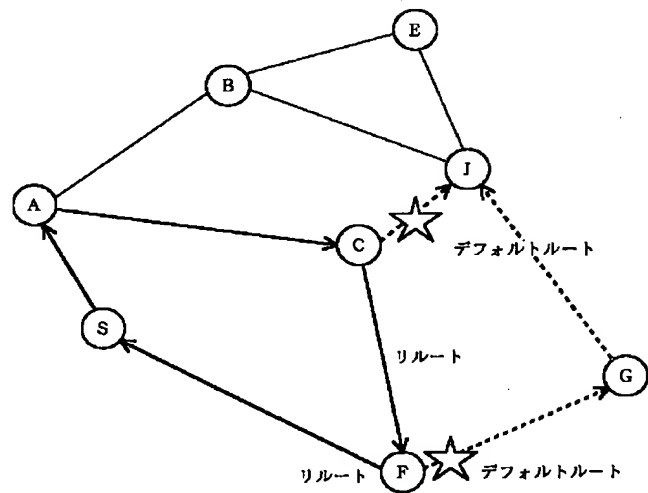
【図 1】



【図 2】

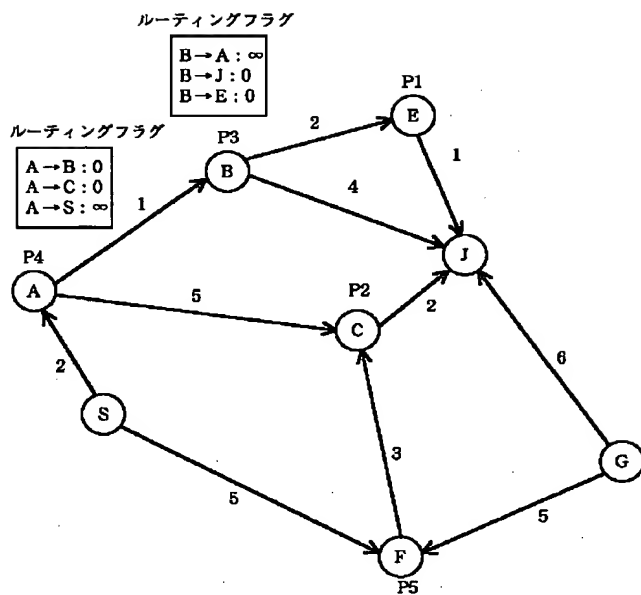


【図 3】

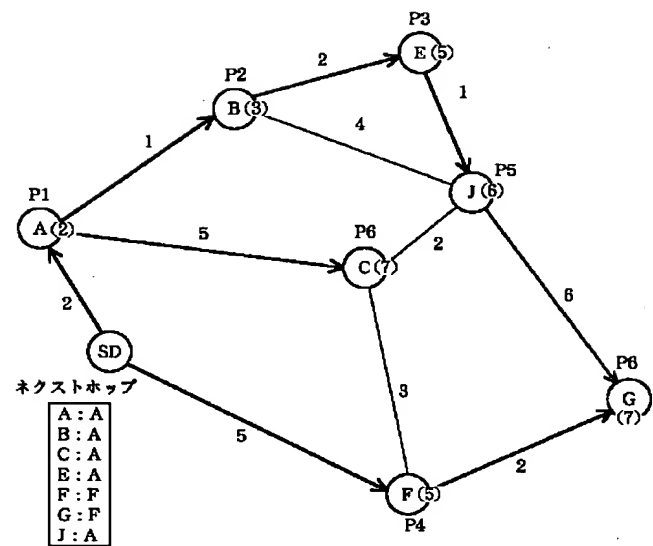




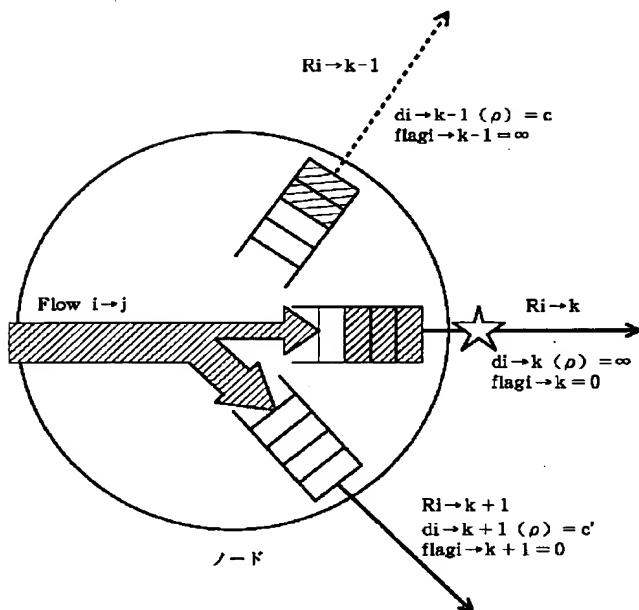
【図 4】



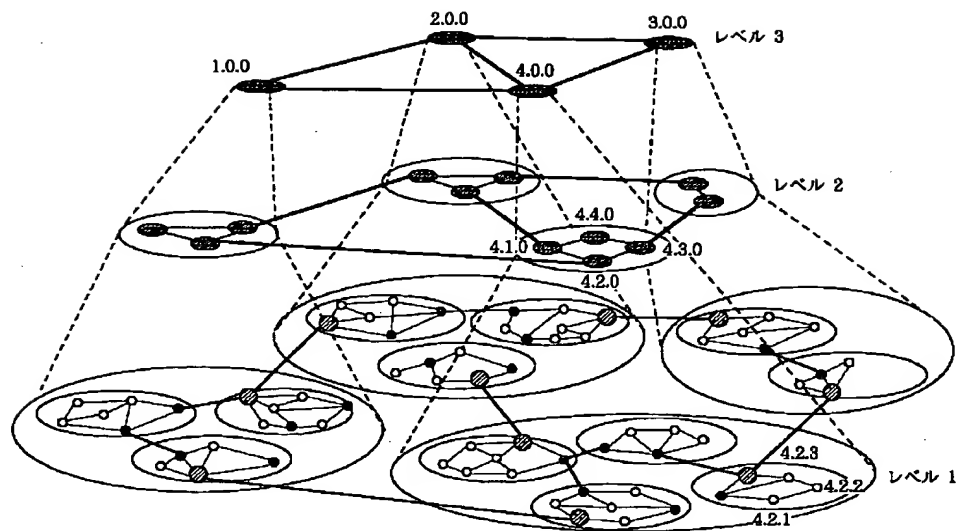
【図 5】



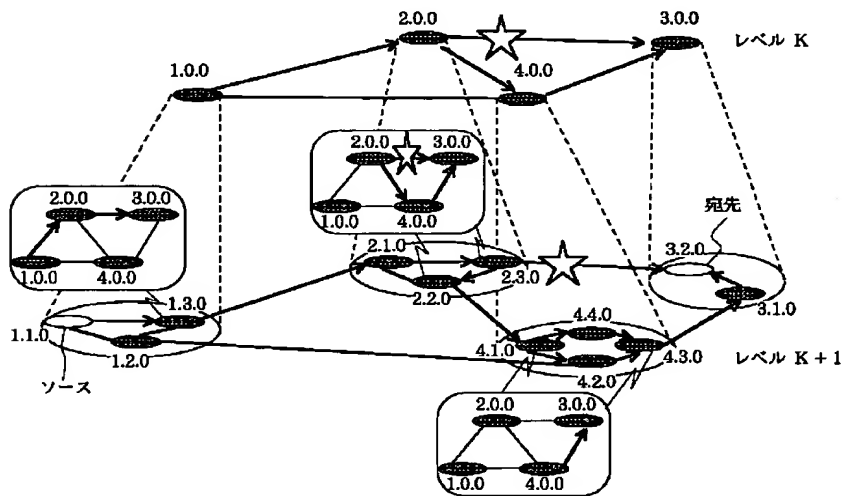
【図 6】



【図 7】



【図 8】



フロントページの続き

Fターム(参考) 5K030 GA13 HC01 JL07 KA01 KA05  
 LB07 LC11 LE03 MA01  
 5K033 AA03 CB06 DA05 DB01 DB12  
 DB14 DB18 EA07 EB02 EC03  
 9A001 CC06 CC07 FF03 GG01 GG06  
 GG19